Proudly Sponsored by:

Microsoft

A CLOUD GURU

vNEXT

Machine Learning + DevOps *using* Azure ML Services

Azure Nights Meetup, Thu 18 Jul 2019

Rolf Tesmer
Microsoft Cloud Architect | Azure | Data | Analytics | AI
Mr. Fox SQL Blog - https://mrfoxsql.wordpress.com/
Linked In - https://www.linkedin.com/in/rolftesmer/

https://mrfoxsql.wordpress.com/2019/06/11/machine-learning-devops-ml-devops-together-at-last/

Todays Agenda

1. What is **DevOps**
2. *The zero!* **Pain Points** in **Data Science projects**
3. *The hero!* **Azure Machine Learning Service**
4. **Demo** - *lets see this puppy in action*
5. **Different ways** to build **ML DevOps** pipelines
6. **Explainability** + **Reproducibility**… *What can I do?*
7. And, *we're clear*… Q&A

*Our strategy is to build best-in-class **platforms** and productivity services for an **intelligent cloud and an intelligent edge** infused with **artificial intelligence** ("AI").*

*With Azure we want to **democratize AI** so every company can be an AI company, and every partner can build AI solutions, while **helping customers build AI capability to transform**.*

Microsoft

What exactly is DevOps? *And Why Should I Care?*

DevOps is a software engineering practice that aims at unifying software development and software operation. The main characteristic of the DevOps movement is to strongly advocate automation and monitoring at all steps of software construction, from integration, testing, releasing to deployment and management.

GOAL: DevOps enables faster time to market, lower failure rate, shortened lead times, automated compliance, release consistency.

method of development → **Agile != DevOps** ← *method of deployment*

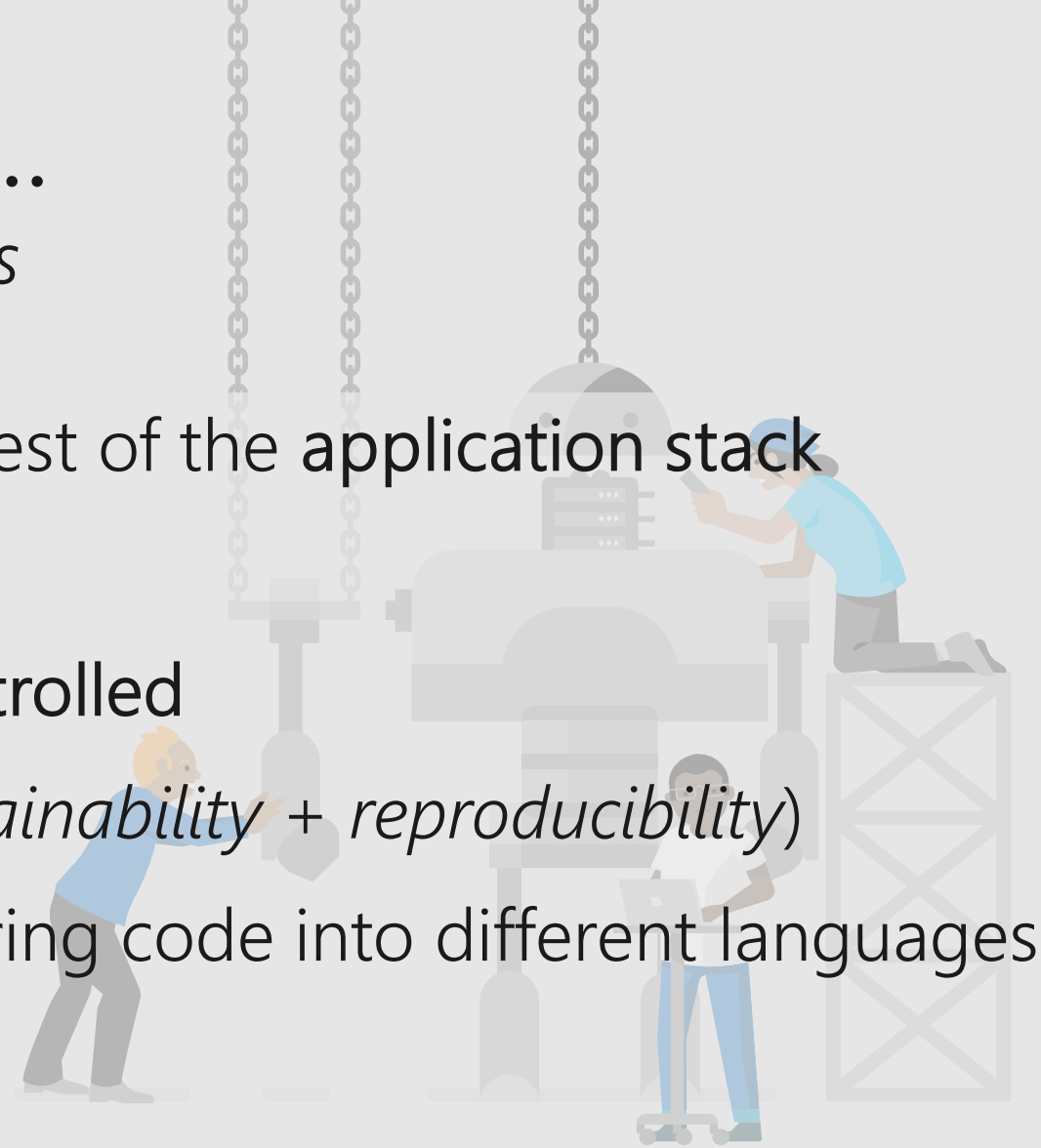
https://en.wikipedia.org/wiki/DevOps

https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/team-data-science-process-for-devops

7x Data Science Pain Points…

…when Applying DevOps Methods / Practices

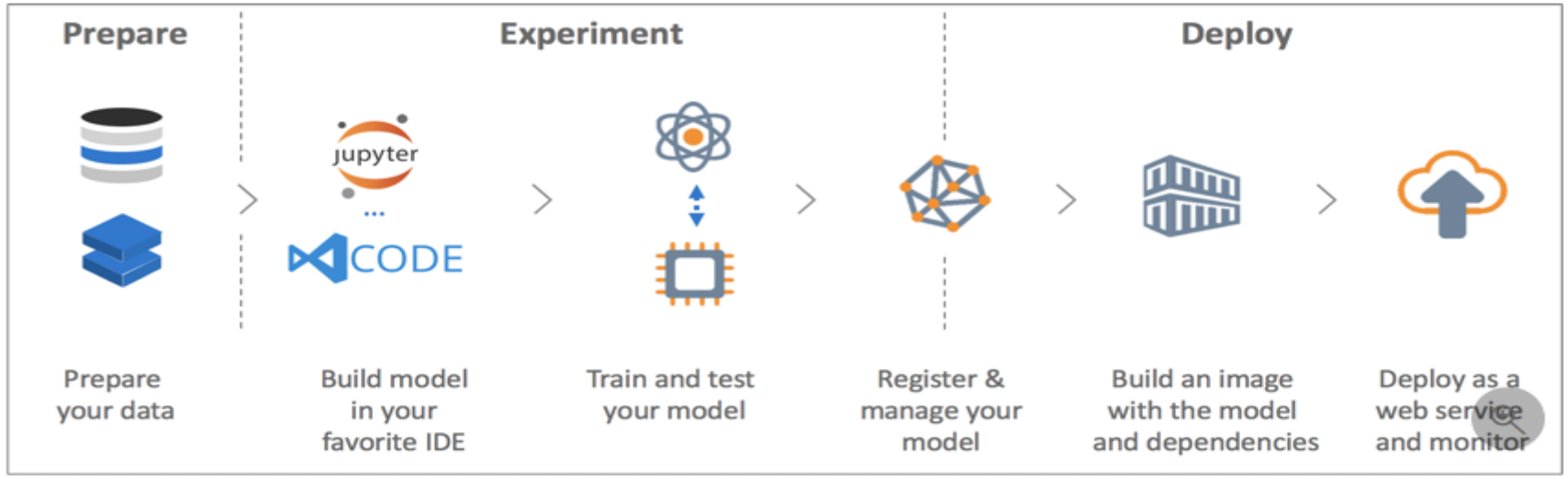
1. **ML stack** might be **different** from rest of the **application stack**
2. Testing **accuracy** of ML model
3. ML code is not always **version controlled**
4. Hard to **reproduce** models (*ie explainability + reproducibility*)
5. Need to **re-write** featurizing + scoring code into different languages
6. Hard to track **breaking changes**
7. Difficult to **monitor** models & determine when to **retrain**



Azure ML Services

https://azure.microsoft.com/en-us/services/machine-learning-service/

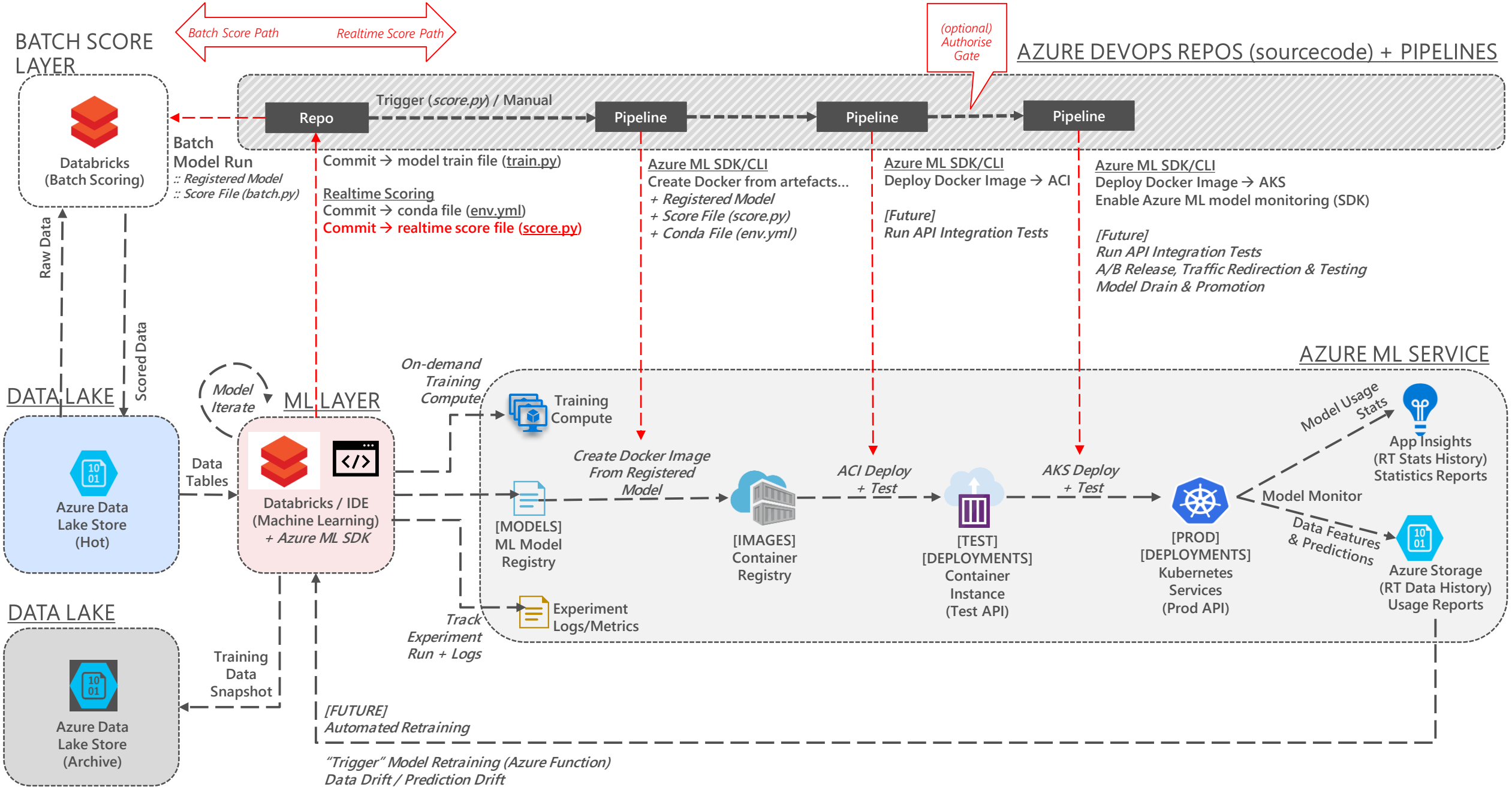
Azure Machine Learning service provides a cloud-based environment you can use to develop, train, test, deploy, manage, and track machine learning models.



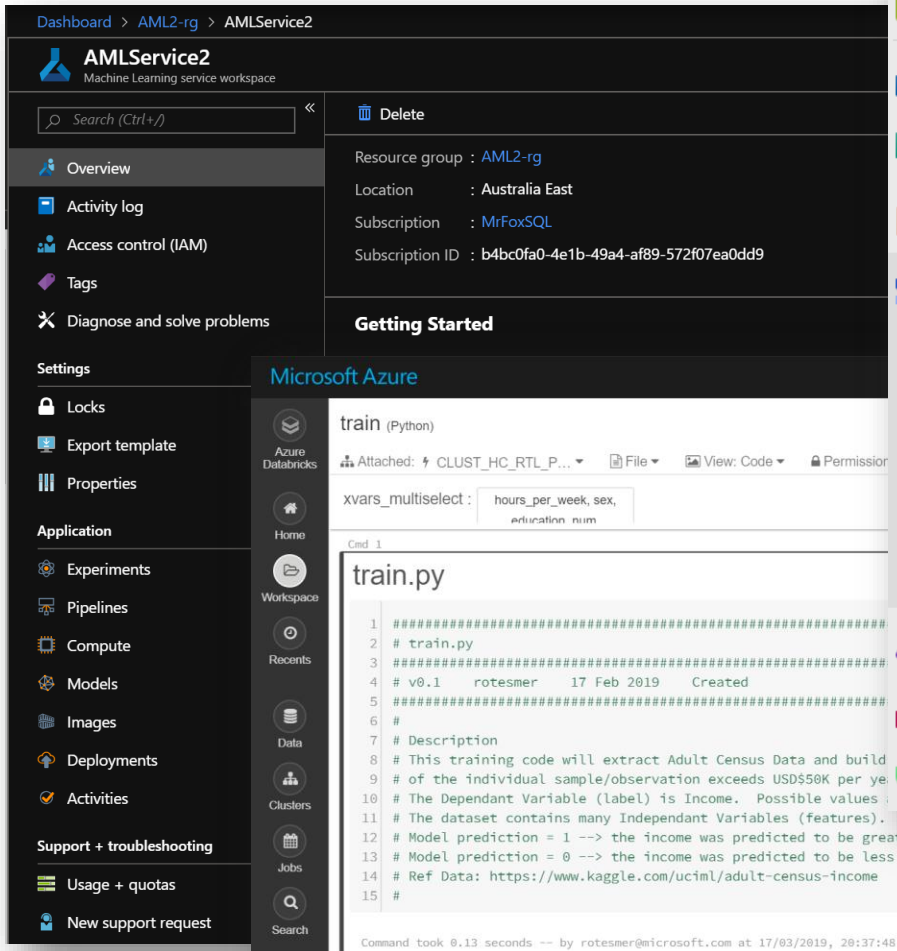
Azure Machine Learning service fully supports open-source technologies. So you can use tens of thousands of open-source Python packages with machine learning components. Examples are TensorFlow and scikit-learn. Support for rich tools makes it easy to interactively explore data, transform it, and then develop and test models. Examples are Jupyter notebooks or the Azure Machine Learning for Visual Studio Code extension. Azure Machine Learning service also includes features that automate model generation and tuning to help you create models with ease, efficiency, and accuracy.

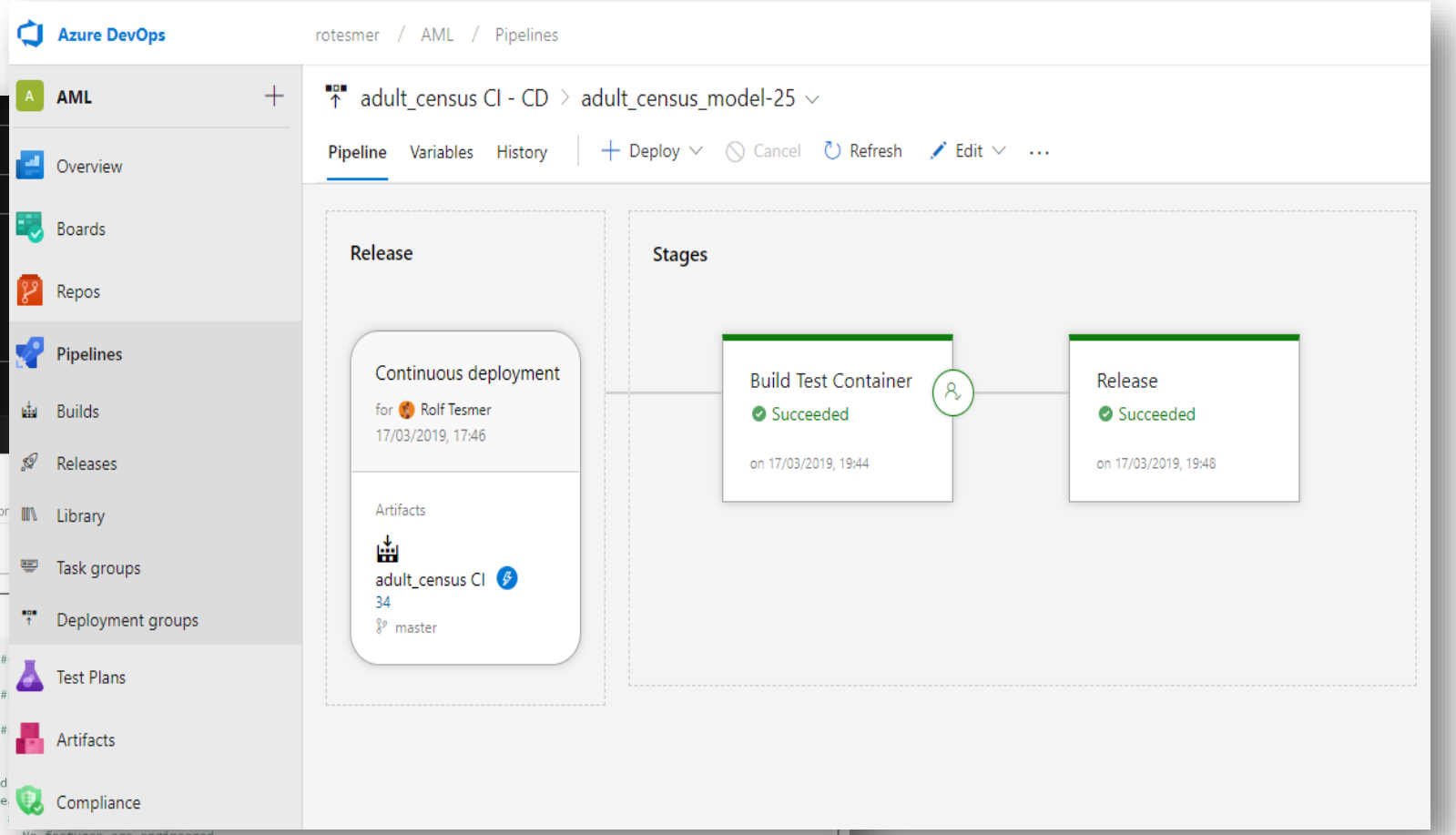
Leverage your favorite frameworks

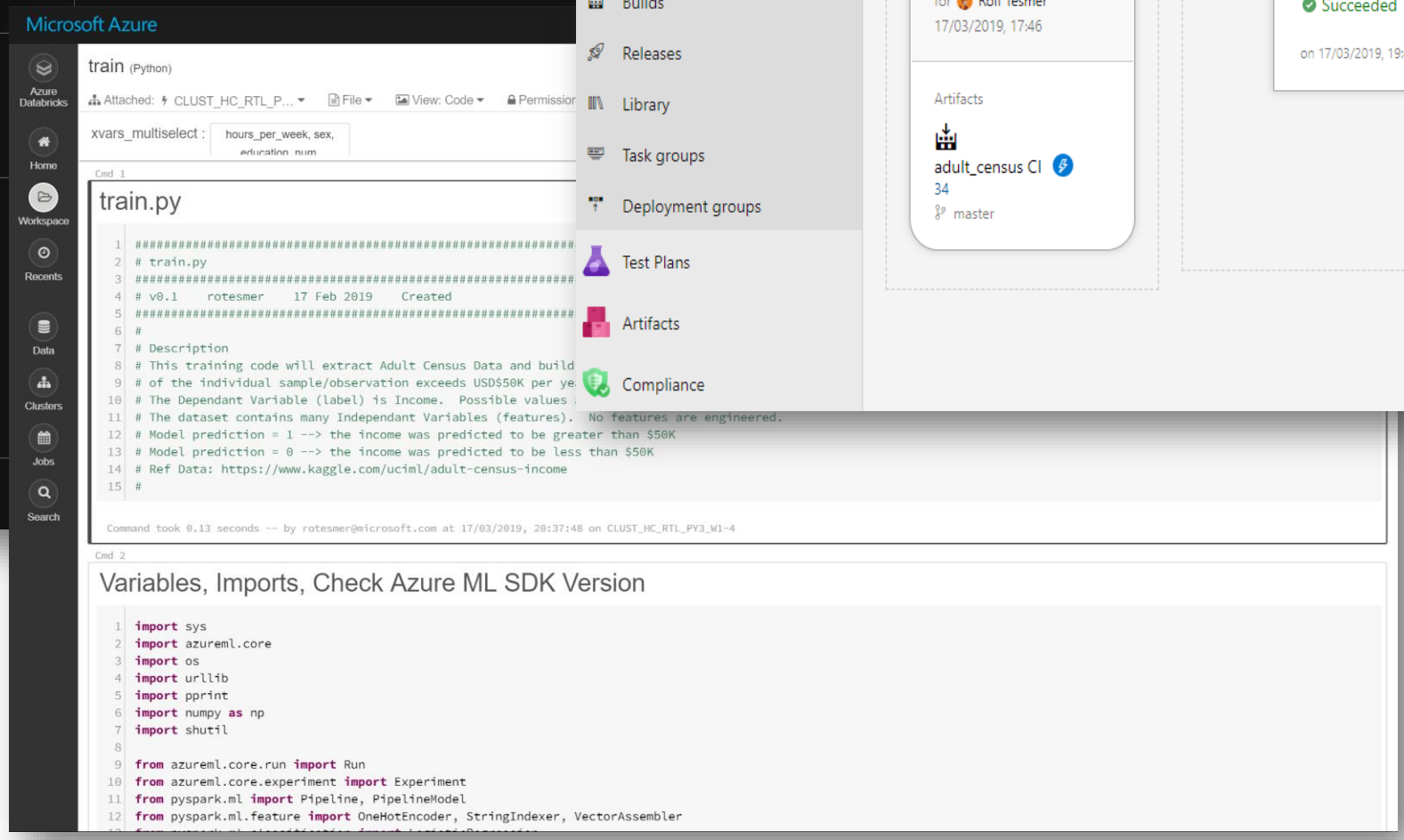
TensorFlow | MS Cognitive Toolkit | PyTorch | Scikit-Learn | ONNX | Caffe2 | MXNet | Chainer


BATCH SCORE LAYER
Batch Score Path
Realtime Score Path
(optional) Authorise Gate
AZURE DEVOPS REPOS (sourcecode) + PIPELINES
Databricks (Batch Scoring)
Repo
Trigger (score.py) / Manual
Pipeline
Pipeline
Pipeline
Batch Model Run
:: Registered Model
:: Score File (batch.py)
Commit → model train file (train.py)
Realtime Scoring
Commit → conda file (env.yml)
Commit → realtime score file (score.py)
Azure ML SDK/CLI
Create Docker from artefacts...
+ Registered Model
+ Score File (score.py)
+ Conda File (env.yml)
Azure ML SDK/CLI
Deploy Docker Image → ACI
[Future]
Run API Integration Tests
Azure ML SDK/CLI
Deploy Docker Image → AKS
Enable Azure ML model monitoring (SDK)
[Future]
Run API Integration Tests
A/B Release, Traffic Redirection & Testing
Model Drain & Promotion
Raw Data
Scored Data
AZURE ML SERVICE
DATA LAKE
Model Iterate
ML LAYER
On-demand Training Compute
Training Compute
Azure Data Lake Store (Hot)
Data Tables
Databricks / IDE (Machine Learning)
+ Azure ML SDK
Create Docker Image From Registered Model
ACI Deploy + Test
AKS Deploy + Test
[MODELS] ML Model Registry
[IMAGES] Container Registry
[TEST] [DEPLOYMENTS] Container Instance (Test API)
[PROD] [DEPLOYMENTS] Kubernetes Services (Prod API)
Model Usage Stats
App Insights (RT Stats History) Statistics Reports
Model Monitor
Data Features & Predictions
Azure Storage (RT Data History) Usage Reports
Track Experiment Run + Logs
Experiment Logs/Metrics
DATA LAKE
Training Data Snapshot
Azure Data Lake Store (Archive)
[FUTURE] Automated Retraining
"Trigger" Model Retraining (Azure Function)
Data Drift / Prediction Drift

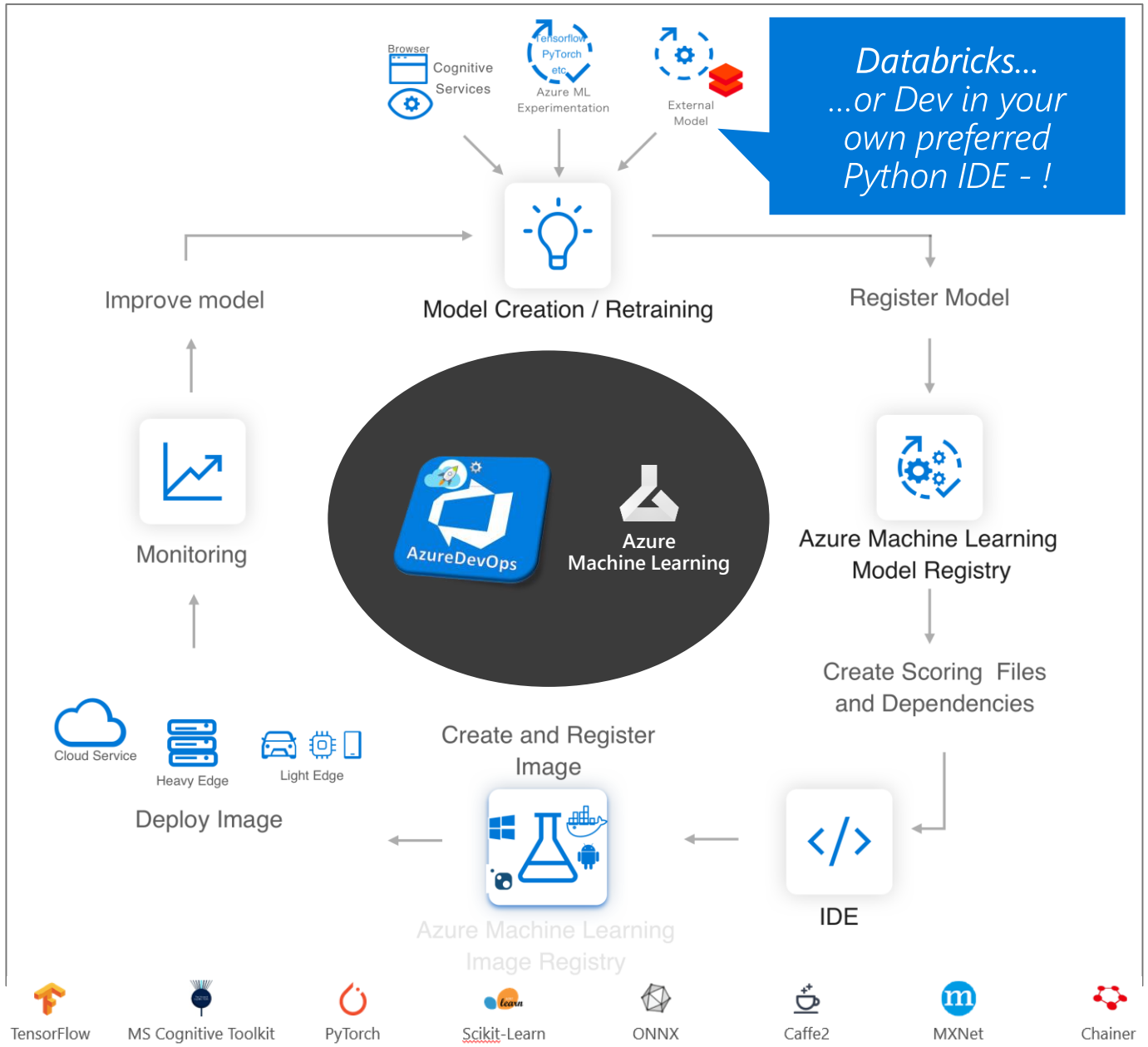
Demonstration







Using the Azure ML Service: *So lets just recap shall we…?*

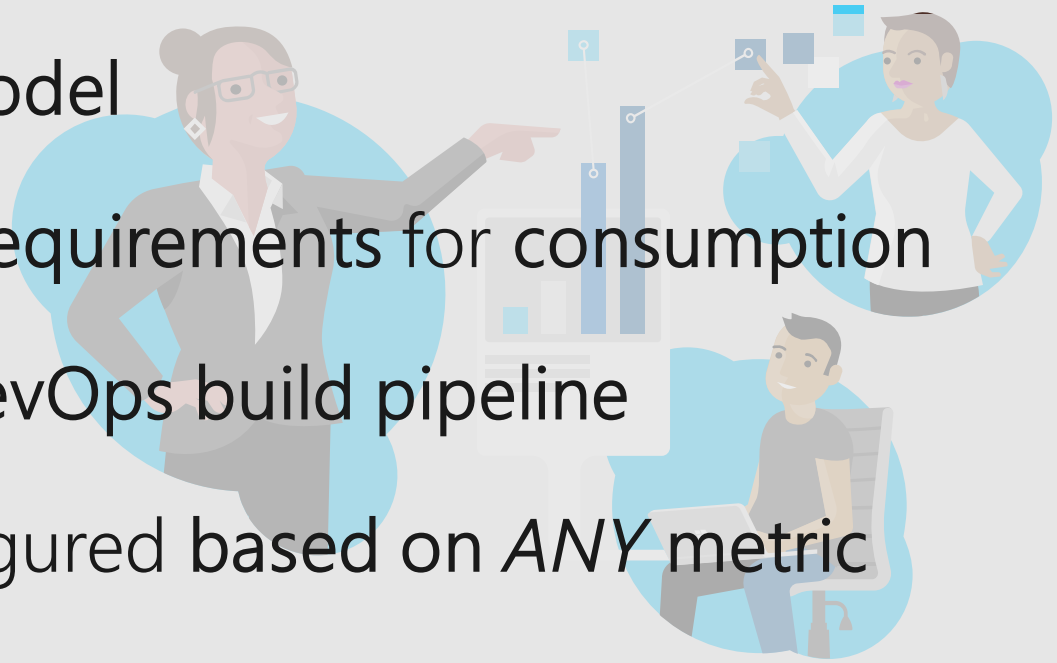


Workflow Steps

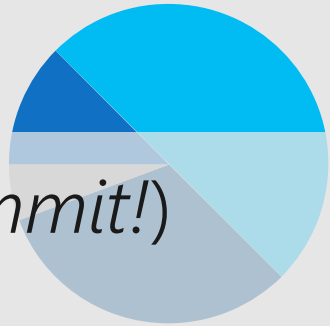
1. Develop ML training scripts in Python (train.py)
2. Create and configure a compute target.
3. Submit the scripts to the configured compute target to run in that environment. During training, the compute target stores run records to a datastore. There the records are saved to an experiment.
4. Query the experiment for logged metrics from the current and past runs. If the metrics do not indicate a desired outcome, loop back to step 1 and iterate on your scripts.
5. Once a satisfactory run is found, register the persisted model in the model registry.
6. Develop a scoring script (score.py)
7. Create a Docker Image and register in image registry.
8. Deploy the image as a web service in Azure.
9. Monitor the deployed Web Service API for drift
10. Trigger an ML model retraining event if required

5x considerations for building + releasing models

1. Different DevOps pipelines **per model**
2. Pipelines should match **business requirements** for **consumption**
3. Put model training into its **own DevOps build pipeline**
4. Retraining **triggers** could be configured **based on *ANY* metric**
5. *Don't forget* about **explainability – build this into DevOps**



5x ways ML DevOps supports explainability



1. continued model code **management + versioning** (*ie commit!*)
2. trained model artefacts **management + versioning** (*ie register!*)
3. **automate** DevOps build + release processes (*ie no click ops!*)
4. training **data retention** (*ie keep all data used to train models*)
5. scoring + prediction **data retention** (*ie model monitoring*).

https://docs.microsoft.com/en-us/azure/machine-learning/service/machine-learning-interpretability-explainability

© 2016 Microsoft Corporation. All rights reserved.

Appendix – References

References

1. https://www.youtube.com/watch?v=nst3UAGpiBA
2. https://github.com/microsoft/MLOpsPython
3. https://docs.microsoft.com/en-au/azure/machine-learning/service/concept-model-management-and-deployment
4. https://mrfoxsql.wordpress.com/2019/06/11/machine-learning-devops-ml-devops-together-at-last/
5. https://docs.microsoft.com/en-us/azure/machine-learning/service/machine-learning-interpretability-explainability
6. https://azure.microsoft.com/en-us/services/machine-learning-service/
7. https://en.wikipedia.org/wiki/DevOps
8. https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/team-data-science-process-for-devops

Azure Databricks

Enable collaboration between data scientists and data engineers with an interactive productive workspace

Prepare and clean data at massive scale with the language of your choice

Build and train models with pre-configured machine learning and deep learning optimized clusters

+

Azure Machine Learning

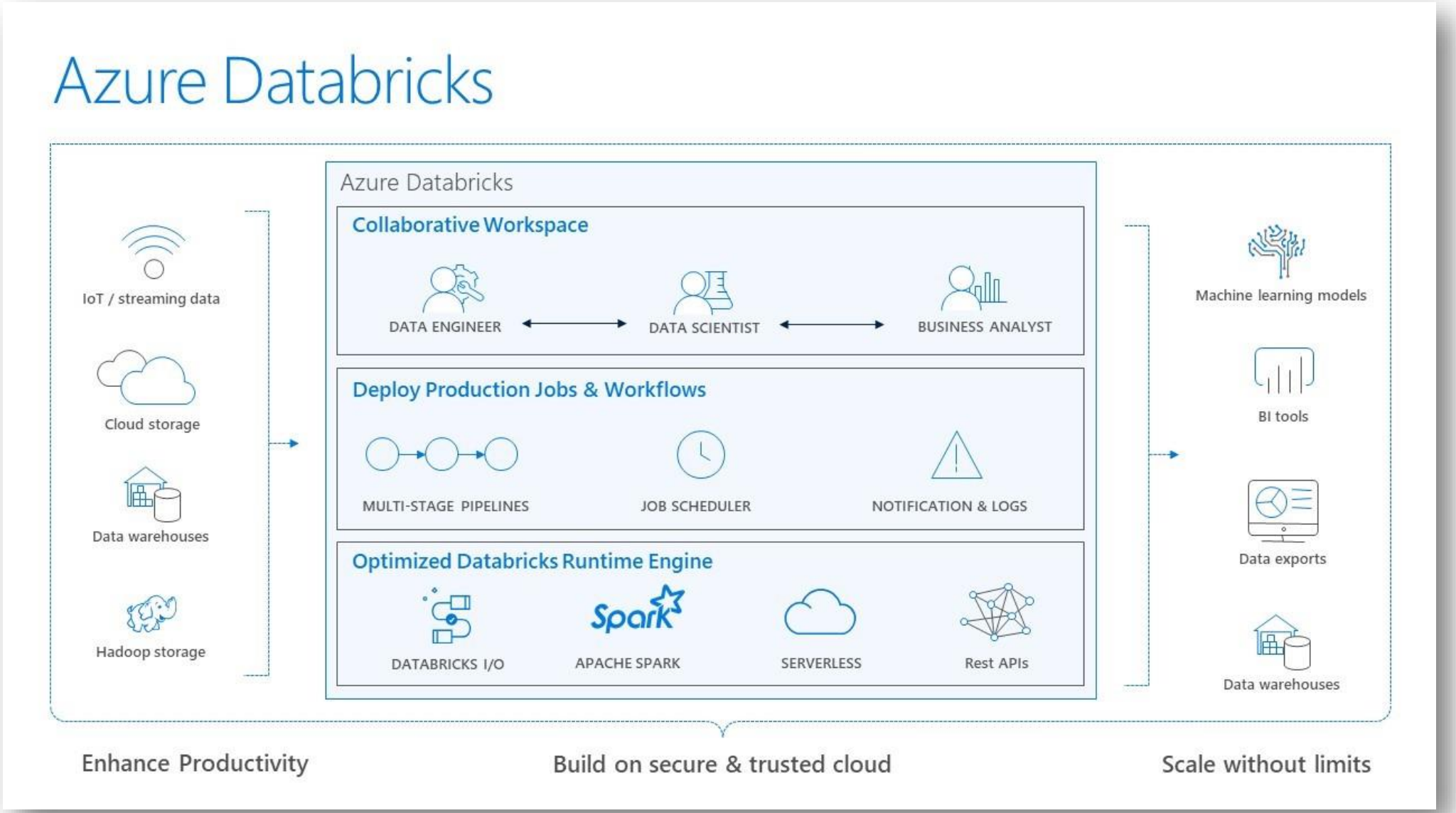
Track experiments for reproducibility and auditing needs.

Identify and promote best performing models into production

Deploy and manage your models using containers to run them anywhere

Azure Databricks

https://azure.microsoft.com/en-us/services/databricks/



Azure DevOps

https://azure.microsoft.com/en-us/solutions/devops/?v=18.44

Introducing Azure DevOps

Azure Dev Ops...
...or use your own preferred DevOps Automation suite of tooling - !

Azure Boards

Plan, track, and discuss work across teams, deliver value to your users faster.

Azure Repos

Unlimited cloud-hosted private Git repos. Collaborative pull requests, advanced file management, and more.

Azure Pipelines

CI/CD that works with any language, platform, and cloud. Connect to GitHub or any Git provider and deploy continuously to any cloud.

Azure Test Plans

The test management and exploratory testing toolkit that lets you ship with confidence.

Azure Artifacts

Create, host, and share packages. Easily add artifacts to CI/CD pipelines.